Exam 1 Review

CS 2308
Fall 2013

Jill Seaman

Exam 1

- Mon, Oct. 7 and Tues, Oct. 8
- In class, closed book, closed notes, clean desk
- 20% of your final grade
- 80 minutes to complete it
- I recommend using a pencil (and eraser)
- All writing will be done on the test paper I will hand out.
- No calculators.

Exam Format

- 100 points total
  - Writing programs/functions/code
  - Multiple choice
  - Fill-in-the-blank/short answer
  - Tracing code (what is the output)
  - Tracing search/sort algorithms

C++ Programming on Linux

- What is Linux
- Linux file system
- Basic shell commands

pwd	more/less/cat
ls	cp
cd	mv
mkdir	rm
rmdir	

- Basic file editing (nano, etc.)
- edit, compile, run

```
nano
g++
./a.out
```

- know how to use the commands

Chapters 1-7 Review

- Know how to program with arrays and functions.

- Passing parameters by reference and by value
- Passing arrays to functions, processing arrays
- Partially filled arrays

- Understand Programming Assignment 1

Ch 11: Structured Data

- Structures:
  - Definition (new data type)
  - Variable definitions
  - How to access members (fields)
  - Operations (which are valid)
  - Arrays of structures
  - Nested structures
  - Structures as function args

Ch.8: Searching and Sorting Arrays

- Searching
  - Linear Search
  - Binary Search
- Sorting
  - Bubble Sort
  - Selection Sort
- Efficiency
  - Growth rate functions: which are faster/slower
  - Efficiency of each searching/sorting algorithm

> You **will not** need to know the code
> --but I may ask you to implement linear search
>
> You **will** need to be able to demonstrate the algorithms
> --see exercises at end

Ch 9: Pointers

- Address operator (&)
- Pointer variables: how to define (data type)
- Dereferencing operator (*)
- Pointers and arrays
  - * an array variable is the address of its first element
  - * array[index] = *(array + index)
- Pointer arithmetic (if ptr points to a var of type d):
  - * ptr + n = address in ptr + n * sizeof(d)
- Initializing Pointers

Ch 9: Pointers, cont.

- Comparing pointers
- Pointers as function parameters
  * Pass by reference using pointers as parameters
  * Pointers used as parameters accepting arrays as arguments
- Dynamic memory allocation
  * new operator
  * new with arrays
  * delete
  * return pointers from functions

Example Programming Problem

Write a function that accepts an array of integers and the size of the array and prints out a table listing how many values in the array fall in each of the following ranges:

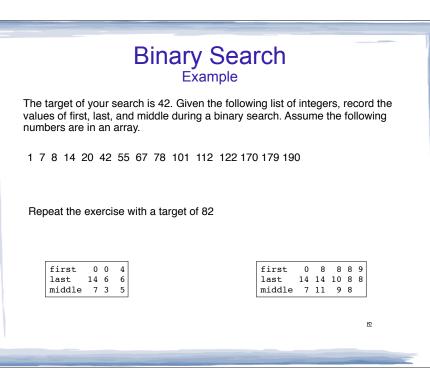less than 50
50 to 59
60 to 69
70 to 84
85 to 99
over 100

Example Tracing Problem

What will the EXACT output of the following program be?

```
int main () {
   int *ptr1, *ptr2;
   int foo1;

   foo1 = 42;
   ptr1 = &foo1;
   *ptr1 = 13;
   ptr2 = ptr1;

   cout << "foo1 - " << foo1 << endl;
   cout << "*ptr1 - " << *ptr1 << endl;
   cout << "*ptr2 - " << *ptr2 << endl;

   int x[] = {1,2,3};
   ptr1 = &x[1];
   *ptr2 = *(x+1);

   cout << endl;
   cout << "*ptr1 - " << *ptr1 << endl;
   cout << "*ptr2 - " << *ptr2 << endl;

}
```

Binary Search
Example

The target of your search is 42. Given the following list of integers, record the values of first, last, and middle during a binary search. Assume the following numbers are in an array.

1 7 8 14 20 42 55 67 78 101 112 122 170 179 190

Repeat the exercise with a target of 82

first	0	0	4
last	14	6	6
middle	7	3	5

first	0	8	8	9
last	14	14	10	8
middle	7	11	9	8

Sorting
Example

Use the following array for both questions:

11	8	14	7	12	18	2	17
0	1	2	3	4	5	6	7

Show the contents of the array after 2 passes of the selection sort

Show the contents of the array after 2 passes of the bubble sort

How to Study

- Review the slides
  * understand all the concepts
- Use the book to help understand the slides
  * there will be no questions over material (or code) that is in the book but not on the slides
- Review programming assignments (fix yours!)
  * view solutions in my office
- Try some exercises from the book
- Practice, practice, practice
- Get some sleep